Hw

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/960,529	09/21/2001	Benjamin Renaud	BEAS-01067US0	5297

23910 7590 11/15/2005

FLIESLER MEYER, LLP
FOUR EMBARCADERO CENTER
SUITE 400
SAN FRANCISCO, CA 94111

EXAMINER	
VU, TUAN A	
ART UNIT	PAPER NUMBER
2193	

DATE MAILED: 11/15/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 09/960,529	**Applicant(s)** RENAUD, BENJAMIN
	Examiner Tuan A. Vu	**Art Unit** 2193

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *06 October 2005*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-9,13-16,19-28,37,38,40-49,55 and 56* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-9,13-16,19-28,37,38,40-49,55 and 56* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

1. This action is responsive to the Application's response filed 10/6/2005.

Claim 39 has been canceled and claims 55-56 newly added. Claims 1-9, 13-16, 19-28, 37, 38, 40-49, 55 and 56 have been resubmitted for examination.

Claim Objections

2. Claims 55-56 are objected to because of the following informalities: there appears to be a missing *hyphen* between 'computer' and 'implemented' at line 1 of claims. Appropriate correction is required.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

4. Claims 1-5, 7, 9, 14-16, 37, and 40-44 are rejected under 35 U.S.C. 102(e) as being anticipated by Knutson, USPN: 6,557,100 (hereinafter Knutson).

As per claim 1, Knutson discloses a method of automatically deploying an application across a distributed computing domain including a plurality of processing devices, the method comprising:

automatically scanning for an undeployed application stored in an application directory accessible (Note: JAR unpacking environment leading to identification of cached or previously deployed EJBs reads on application directory – see col. 5, lines 35-54; col. 4, lines 46-48; Fig. 5 – because the JAR *515* and JAR *520* enable application data storage being archived to be unpacked and accessible) to at least one of the plurality of processing devices (e.g. Fig. 6; Fig. 7; col. 4, lines 38-41 – Note: identification of applications that need to be redeployed is equivalent to scanning for previously deployed application, accessible to at least one of the client machines);

recognizing an undeployed application in the application directory (e.g. Fig. 6,7; col. 4, lines 38-41 – Note: generating of new EJB when checking on cached of previous reads on recognizing of an undeployed application in the JAR); and

deploying the undeployed application to a selected portion of the plurality of processing devices (Note: Knutson discloses selected portion of the plurality of processing devices via the session and protocol via which a number of specific machines -- those in the LAN which submit the request for a bean -- must communicate to obtain the EJB deployment – see col. 3, lines 32-67; *filter, rules* – col. 3, lines 60-67; *security rules, Session beans* – col. 4, line 42 to col. 5, line 3; *scalability* - col. 1, line 33 to col. 2, line 12 – the specificity of a client per client and session associated thereto reads on selected ones of the plurality of processing devices) such that the applications is capable of being executed by said selected portion.

As per claim 2, Knutson discloses the steps of:

obtaining a list of applications stored in the application directory (e.g. *cached* -col. 2, line 40-45 – Note: descriptor entries --from JAR being unraveled -- being organized in cache are equivalent to list);

comparing the list of applications stored in the application directory to a list of previously deployed applications in order to select the application to be deployed, and deploying the selected application to the selected portion of the plurality of processing devices (e.g. col. 5,lines

42-51; Fig. 7 – Note: comparing with previously deployed descriptor being cached is equivalent to comparing against list of application stored in application directory).

As per claim 3, see Knutson (Fig. 7; col. 5 line 42 to col. 6, line 5; col. 4, lines 38-41 – Note: for any change identified from comparing with cached list, a new version of file is created, and this reads on an application being absent from the previously deployed list because a newer version is not in the currently deployed list)

As per claims 4 and 5, see Knutson (e.g. *descriptor* - Fig. 7; col. 5 line 42 to col. 6, line 5 – Note: descriptor detected from parsing a archive structure is equivalent to attribute of a file containing bean component)

As per claim 7, Knutson discloses indicator being attribute of a file associated with a file containing the application (Fig. 7; *data which represents* -- col. 5, line 25 – Note: EJB descriptor is attribute of bean contained in JAR file)

As per claim 9, Knutson discloses analysis of attributes from undeployed application (e.g. step 725 – Fig. 7- Note: descriptor identified as not being cached or different from a previously cached descriptor is attribute of undeployed application) and attributes from distributed computing domain (cached descriptor or JAR- step 735 Fig. 7 – Note: redeploying of application to a selected portion of clients has been disclosed in view of rationale in claim 1).

As per claim 14, Knutson discloses redeployment of application containing plurality of EJBs (e.g. col. 5, lines 35-41), hence has disclosed undeployed application beans being contained in a single file.

As per claim 15, Knutson discloses beans being separate files (Fig. 3; *class files* - col. 4, lines 44-48 – Note: class identified from a Jar are different files).

As per claim 16, see Fig. 1B.

As per claim 37, this is a computer medium claims with medium to embody instructions for performing the method claim 1, which Knutson also discloses (see Knutson: col. 6, lines 19-33).

As per claim 40, Knutson discloses a processing system including a first processing device, a memory accessible by the first processing device, the processing system comprising:

a group of processor readable instructions stored in the memory device and operating the first processing device to perform a group of steps:

automatically (scanning for an undeployed application) stored in an application directory accessible to first processing device,

recognizing (undeployed application) in the application directory; and

deploying (undeployed application to a selected portion of the processing system); all of which steps having been addressed in claim 1.

As per claim 41, Knutson discloses the selected portion of the processing system includes the first processing device (server *102* – Fig. 1A; col. 4, lines 49 to col. 5, line 7; col. 5, line 55 to col. 6, line 11 – Note: deployment of beans via recompiling effected by the server discloses selected portion including first processing device, e.g. compiling capabilities of server machine)

As per claim 42, Knutson discloses including a second processing device in communication with the first processing device, wherein the selected portion of the processing system includes the second processing device (e.g. Fig. 6,7 – Note: the JAR scanning and selection of application files to be redeployed– or first processing device- in conjunction or communication with the redeployment process, i.e. a second processing device, itself in the same environment, is equivalent to selected portion including a second processing device).

As per claim 43, in view of claim 42, where the selecting for deployment and the deployment process is executed on the same server machine, Knutson has disclosed the first processing device and the second processing device are located on a first computer.

As per claim 44, Knutson discloses the first processing device is located on a first computer and the second processing device is located on a second computer (computer *108, 110, 112* – Fig. 1A).

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 6, 8, 19-28, 38, 45-49, and 55-56 are rejected under 35 U.S.C. 103(a) as being unpatentable over Knutson, USPN: 6,557,100.

As per claim 6, Knutson does not explicitly discloses a file date as attribute; but official notice is taken that versioning of a file (see Knutson: col. 5, lines 31-34) or caching of a filename with incorporating or including a date attribute/descriptor (as in *deployment descriptor* by Knutson – Fig. 7) therein was a well known concept at the time the invention was made. In view of Knutson updating of a version and checking of descriptor (col. 5, lines 26-34; Fig. 7), the limitation to making a attribute date is implicitly disclosed or would have been obvious because incorporating a date as attribute for versioning file enables clear distinguishing of versions using a time base, a concept universally known as non-repetitive or un-duplicable.

As per claim 8, in view of the rationale for obviousness regarding the identification a file version by attribute using system date as set forward in claim 6, the setting of a EJB identifier using a date attribute would also have been obvious according to known concept as set forth above.

As per claim 19, Knutson discloses a method for automatically maintaining an application object across a distributed computing domain, the object contained within one application file, and said computing domain including a plurality of processing devices, the method comprising the steps:

retrieving a list of all of the application files located within an application directory (e.g. steps *600-615*- Fig. 6; steps *700-715* – Fig. 7; col. 4, lines 38-41 – Note: application directory is met by JAR and JAR *515, 520* of Fig. 5 and related unpacking environment - see col. 5, lines 35-54; col. 4, lines 46-48; Fig. 5);

comparing the list of all of the files located within an application directory to a list of all of the files associated with previously deployed application objects(e.g. col. 5, lines 42-62; col. 4, lines 38-41);

for each application file, deploying the application object contained in the application file when the application file is absent from the list of all the files associated with previously deployed application objects (e.g. col. 5, line 47 to col. 6, line 5; col. 4, lines 38-41 – Note: generating of new EJB when checking on cached of previous reads on recognizing of un undeployed application in the JAR and deploying it – see Fig. 7) for each application file, redeploying the application object contained in the application file when the application file differs from the corresponding file on the list of all of the files associated with previously deployed application objects (e.g. col. 5, lines 42-67 -- Note: generating of new EJB when checking on cached of previous reads on recognizing of undeployed application in the JAR and redeploying it – see *if the EJB is redeployed* – col. 2, lines 44-48).

Knutson does not explicitly disclose that for each application file on the list of all of the files associated with previously deployed application objects, undeploying the application object associated with an application file when the application file on the list of all of the files associated with previously deployed application objects is absent from the list of all of the application files located within the application directory. But in view of the creation of new files as a result of descriptor comparison mismatch, the suggestion as to download or transmit the latest compiled application bean to the user also entails the use of the latest compiled bean and activation of such bean at the client processor. As a result, the concept of undeploying an older version at the client machine is suggested. Hence, it would have been obvious for one skill in the art at the time the invention was made to undeploy any application file being previously used at the client end which is unmatched against the application JAR list files so that the resources can be directed to using the new created file being compiled at the server deployment directory.

As per claim 20, Knutson discloses difference from comparing the value of a deployment indicator associated with an application file with the value of a deployment indicator recorded on the list of previously deployed application objects (see Fig. 6,7).

As per claims 21-26, these claims correspond to claims 5-8, 13, 16, respectively; hence are rejected using the corresponding rejection as set forth therein.

As per claims 27 and 28, see Knutson: Fig. 1A-B.

As per claim 38, this is a computer medium claims with medium to embody instructions for performing the method claims 1, 19 respectively, which Knutson also discloses (see Knutson: col. 6, lines 19-33).

As per claim 45, this is a processing system version claim including processor readable instructions stored in the memory device and operating the first processing device to perform a group of steps as recited in method claim 19 including the steps

retrieving a list of all of the application files;

comparing the list;

for each application file, deploying;

for each application file, redeploying; and

for each application file on the list of all of the files associated

with previously deployed application objects, undeploying;

as recited in claim 19.

Hence this claim is rejected with the corresponding rejection as set forth in claim 19.

As per claims 46-49, these claims correspond to claims 41-44, respectively; hence are rejected using the corresponding rejection as set forth therein.

As per claim 55, Knutson discloses a computer-implemented method for deploying applications to an application server (e.g. Fig. 1A-B; Fig. 3) comprising automatically deploying an application to an application server when corresponding unpackaged application files are added to a smart directory (col. 5, line 47 to col. 6, line 5; col. 4, lines 38-41; col. 5, lines 42-67 -- Note: JAR unpacking environment leading to identification of cached or previously deployed EJBs so to intelligent redeploy the undeployed beans reads on smart application directory).

Knutson does not explicitly disclose automatically undeploying the application to an application server when corresponding unpackaged application files are removed to a smart directory; but this undeploying limitation has been addressed in claim 19 above; and herein would have been obvious for the same reasons as set forth therein.

As per claim 56, see Knutson (col. 5, lines 20-21; step 750 -Fig. 7)

7. Claims 13 is rejected under 35 U.S.C. 103(a) as being unpatentable over Knutson, USPN: 6,557,100; further in view of Seidman et al., USPubN: 2003/0005166 (hereinafter Seidman).

As per claim 13, the technique of setting a server deployment environment with a directory for developing or processing a file as taught by Knutson was a known concept in the server deployment technology and is further evidenced by Seidman. In a method to deploy a bean application similar to Knutson, Seidman discloses a bean deploying system where the JAR files are stored in directory particularly associated with bean identification/name (e.g. pg. 7, para 0106-0108). Further, Seidman discloses a automated schedule for synchronization of data (pg. 2, para 0022; pg. 9, claim 20). This is evidence that enterprise business bounding client machines and server and being equipped with automated program for enforcing synchronizing (see Knutson: col. 5, lines 31-34) of versioned application or data between server persistent storage and client local storage was a known concept at the time the invention was made. Hence, since Knutson also provides a form of synchronizing service so to update versions being in use by client applications with a newer version of applications, it would have been obvious for one skill in the art at the time the invention was made to implement the automated update service operable on time interval as suggested by Seidman to Knutson's service, because that way enterprise business data or application program would be constantly in sync with the persistent storage as suggested by the approach by Seidman, such synchronizing enabling more secure or fault-free operating system or application level within the executing resources of the enterprise network devices.

Response to Arguments

8. Applicant's arguments filed 10/6/2005 have been fully considered but they are not persuasive. Following are the Examiner's observation in regard thereto.

Rejection under 35 USC §103(a):

(A) Applicant has submitted that Examiner is 'confusing deployment of EJBs … runtime behavior' and that the rules in Knutson only deal with governing EJB operations not deployment (Appl. Rmrks, pg. 9, bottom) and that Applicants fail to see how the network/firewall/proxies rules as applied are relevant with selecting processing devices for deployment of applications (Appl. Rmrks, pg. 10). The rejection has now pointed out what in Knutson is analogized to 'selected portion of the plurality of processing devices', rendering the firewall rules argument moot. The rejection has addressed what portions of Knutson have met the deploying of undeployed application in light of scanning package in a directory where package are loaded in order to effect the redeployment of the beans that are deemed to be redeployed based on some criteria. The claim does not enable a clear teaching from reciting of the phrase 'deploying' so to enforce a distinguishing angle that would preclude the JAR/EJB process by Knutson (see Fig. 6-7) from not reading on 'deploying' or 'redeploying'. Particularly, there is no such understanding from Knutson's teaching that would necessarily equate Knutson's EJB deploying process as being a mere 'runtime behavior' (which is a very limited subconcept pertaining to just one executing thread/process within an environment like that of Knutson) as asserted (Appl. Rmrks, pg. 9, bottom), even though, for the sake of argument, 'runtime behavior' can encompass not only such redeployment process and/or 'deploying', as claimed, of EJB/file. Hence, the arguments about runtime behavior are misplaced. Concerning the rebut against inherency of the 'selecting' limitation via the propounding of proxies/firewall rules by the Examiner, first, the arguments are now moot in view of the new grounds of rejection; second, Applicants' proffering of the term 'selecting' appear to have implicated a new limitation when the claim only recites 'to a selected portion'. For the latter, In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., *selecting* - portion of plurality of processors) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

(B) Applicant has submitted that Examiner's view that Seidman processing device is equivalent to 'application directory' cannot provide grounds to support how such directory is capable to execute an application (Appl. Rmrks, pg. 11, middle). The recited 'application directory' does not enforce the interpretation that it is a special application that actually execute anything. The rejection has shown that this application directory limitation is met by the environment wherein cached data is scanned and packages are loaded for deriving decision as to whether or not to redeploy unpackaged files or previously deployed files; or to make new EJB files based on the information obtained from such scanning and unpackaging processes. A directory is a broad term to mean an environment wherein some plurality of items (like files) are listed or stored in order for an application to fetch, retrieve data therefrom, or to manipulate data in order to yield a application result or to persist further data; and Knutson's workplace for JAR unpackaging or packaging – list of file or information entries being retrieved and analyzed - as based on cached application information reads on a directory used for an application, thus 'application directory'. And concerning the concern that Applicants fail to see how scanning a 'package … descriptor … updated package' is same as 'deploying the application object … when the application file is absent … previously deployed application objects' (Appl. Rmrks, pg. 12, top), the rejection has analogized application file as EJBs and list of applications or files as the unpacked JAR with BEANs attributes helping, for example, the identification of previously deployed beans or presence of update versions to meet. Denouncing that certain Knutson's feature does not read on certain claimed features does not amount to helping the Examiner grapple what specifically distinguishes the Knutson reference from the claimed features. Applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references.

(C) Applicant has submitted that neither Knutson nor Seidman as combined in a 103(a) teach or suggest 'deploying the undeployed application to a selected … plurality of processing devices' (Appl. Rmrks, pg. 12-14). The arguments are now moot in view of the new grounds of rejection; and the points raised concerning 'selected portion of the plurality…' and 'application file is absent from the list…" fall under the ambit of the Examiner's observations made in section A and B above.

The claims will stand rejected as set forth in the Office Action.

Conclusion

9. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (272) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (571)272-3719.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 (for non-official correspondence – please consult Examiner before using) or 571-273-8300 (for official correspondence) or redirected to customer service at 571-272-3609.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

VAT

November 8, 2005

TODD INGBERG
PRIMARY EXAMINER